**DREAM Challenge 2022 Predicting gene expression using millions of random promoter sequences by**

**Metformin-121**

***Abstract***

Previous studies of prediction of gene expression usually used lots of methods of data preprocessing based on biological knowledge. However, these methodologies may cost lots of time to design and train.  
Here, we aimed to propose an end-to-end solution of deep learning for predicting gene expression. We developed a novel method to predict gene expression with minimal methods of data preprocessing with bidirectional gated recurrent unit.

**1. Description of data usage**

The provided data were divided into training set (n = 6,000,000) and validation set (n = 739,258) according to row order. By using the one-hot encoding, each character in the DNA sequence is represented by 4 channels ('T':[1,0,0,0], 'C':[0,1,0,0], 'G':[0,0,1,0], 'A':[0,0,0,1],'N':[1,1,1,1]). One-hot encoding in each DNA sequence, including the adapters that flank the promoter and itself, is stored in a list of length 142. The one-hot encoding [0, 0, 0, 0] represented to the missing value was filled in the front end of the list when the length of the DNA sequence was shorter than 142. The predicted value is obtained by dividing the original value by 2 times the median of the data (2 \* np.median(Y)). There are no data generators or data augmentation strategies.

**2. Description of the model**

Our model was built on a combined architecture of 8 bidirectional gated recurrent unit (GRU), an attention layer (Schuster and Paliwal, 1997; Yang et al., 2016), and finally a dense layer. Between attention layer and dense layer, to avoid overfitting, batch normalization is applied. LeakyReLU activation function, a leaky version of Rectified Linear Unit, was used after batch normalization. After LeakyReLU activation, we utilized dropout layer for generalization.

**3. Training procedure**

Our classification training was carried out using mean squared error loss function and ADAM optimizer. Initial learning rate is 0.001.

For the best model, coefficient of determination of training set is 0.5728 and coefficient of determination of validation set is 0.5372.

**4. Other important features**

Models were evaluated on their performance on the validation set for 30 training epochs. The best model was saved based on the lowest loss.

**5. Contributions and Acknowledgement**

**5.1 Contributions**

|  |  |  |
| --- | --- | --- |
| Name | Affiliation | Email |
| Tsai-Min Chen | (1) Graduate Program of Data Science, National Taiwan University and Academia Sinica, Taipei, Taiwan  (2) Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan | b99612040@ntu.edu.tw |
| Chih-Han Huang | ANIWARE, Taipei, Taiwan | robin.ch.huang@gmail.com |
| Hsuan-Kai Wang | Independent Researcher | wanghsuankai@gmail.com |
| Edward S.C. Shih | Institute of Biomedical Sciences, Academia Sinica, Taipei, Taiwan | shihds@gate.sinica.edu.tw |
| Sz-Hau Chen | Development Center for Biotechnology, Taipei, Taiwan | szhau.chentw@gmail.com |
| Chih-Hsun Wu | Artificial Intelligence and E-learning Center, National Chengchi University, Taipei, Taiwan | j20031214@gmail.com |
| Jhih-Yu Chen | Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan | a402250025@gmail.com |
| Kuei-Lin Huang | School of Medicine, China Medical University, Taichung, Taiwan | judyhuang4705@gmail.com |

**5.2 Acknowledgement**

We thankfully acknowledge the DREAM Challenge 2022 Predicting gene expression using millions of random promoter sequences and Dr. Carl de Boer for their help and for making the data publicly available.

**6. References**

**7. Brief instructions on how to run the code**

We suggest creating an empty environment and install the packages by:

pip install -r requirements.txt

The codes of data preprocessing, training, and testing includes in Meformin-121.ipynb. First, put the training data “train\_sequences.txt” and testing data “test\_sequences.txt” into the same folder as Meformin-121.ipynb. After Executing the codes from the top to the bottom, expression\_f4\_b32\_full\_gru\_8\_coeff\_determination\_loss would be the best model. “test\_sequences\_simon\_22.txt” would be the prediction results.